Quantum cryptography - the BB84 protocol

Finn Hoffmann

January 17, 2025

1 Introduction

The need for secure communication has driven the evolution of cryptographic techniques over centuries. Classical cryptography relies on the computational difficulty of specific mathematical problems, such as factoring large integers or solving discrete logarithms. However, the emergence of quantum computers threatens the security of many widely used cryptographic protocols, as algorithms like Shor's algorithm could efficiently solve these problems. Quantum cryptography, on the other hand, leverages the principles of quantum mechanics to ensure security, fundamentally shifting the approach to secure communication.
One of the pioneering protocols in this domain is the BB84 quantum key distribution (QKD) protocol [1]. Unlike classical cryptography, which depends on computational hardness, BB84 provides information-theoretic security, meaning its security is guaranteed by the laws of physics rather than computational assumptions.
The BB84 protocol enables two parties, typically referred to as Alice and Bob, to securely generate a shared secret key over an insecure channel, even in the presence of an eavesdropper (Eve). The protocol consists of the following key steps:

- Quantum Transmission: Alice prepares a sequence of qubits in one of two mutually unbiased bases (rectilinear or diagonal) and sends them to Bob over a quantum channel. Each basis is randomly chosen, and the qubit states encode the binary values 0 or 1.
- Measurement by Bob: Upon receiving the qubits, Bob measures each one in a randomly chosen basis, either rectilinear or diagonal. Due to the no-cloning theorem, if Bob's measurement basis matches Alice's preparation basis, he obtains the same bit value. If not, the outcome is probabilistic.
- Basis Reconciliation: Alice and Bob communicate over a classical channel to compare the bases they used for each qubit without revealing the actual bit values. They retain only the bits where their bases matched, discarding the rest.
- Error Detection and Privacy Amplification: By publicly comparing a subset of their retained bits, Alice and Bob can estimate the error rate in their key. A high error rate indicates the presence of an eavesdropper. If the error rate is below a certain threshold, they proceed to apply error correction and privacy amplification techniques to distill a final, secure key.

The BB84 protocol is robust against eavesdropping because any attempt by Eve to intercept or measure the qubits will introduce detectable errors in the key. This feature makes BB84 a cornerstone of quantum cryptography and a critical step toward practical quantum-secure communication systems.
In this work we implement a simulation of the protocol to analyze its security and performance against an eavesdropper (implementation available on GitHub [2]).

2 Methods

2.1 Implementation of the BB84 Protocol

The BB84 protocol in this project is implemented using Python and Qiskit, with modular scripts representing the roles of Alice, Bob, and Eve. The implementation simulates key

distribution, measurement, eavesdropping, and error analysis, adhering to the protocol's theoretical framework.

2.1.1 Quantum State Preparation by Alice

Alice's preparation of quantum states is implemented in `alice.py`. A function generates random bit values and measurement bases for n qubits. Depending on the chosen bases, the quantum states are initialized using Qiskit's gate operations, such as Pauli-X and Hadamard gates. These qubits are then transmitted to Bob.

2.1.2 Measurement by Bob

In `bob.py`, Bob simulates the reception of qubits and their measurement. He randomly selects measurement bases for each qubit and applies the corresponding operations. The qubits are measured using a quantum simulator backend from Qiskit. The measurement results yield Bob's sequence of bits and bases.

2.1.3 Eavesdropping by Eve

The optional eavesdropping module in `eav.py` allows Eve to intercept the qubits. Eve performs measurements in randomly chosen bases, which introduces detectable errors if her basis does not match Alice's. This simulates the impact of a potential eavesdropper on the key exchange process.

2.1.4 Error Detection and Key Reconciliation

Utilities for error detection, matching basis indices, and private key extraction are implemented in `utils.py`. These functions compare Alice's and Bob's measurement results, identify matching bases, and generate the shared key. If eavesdropping is detected through a high error rate, the protocol is aborted.

2.1.5 Statistical Analysis and Simulation

The `simulate.py` module facilitates repeated simulations of the protocol to analyze its performance. Key metrics such as the potential and actual bit-switch rates, as well as the error rate, are calculated. These simulations provide insights into the protocol's robustness under various conditions.

2.2 Simulation Setup

The main simulation script, `main.py`, integrates all components to execute the BB84 protocol. Parameters such as the number of qubits (n_qubits) and the eavesdropping toggle can be configured to test different scenarios.

3 Theoretical Aspects and Analysis of BB84

3.1 Theoretical Security of BB84

The BB84 protocol is grounded in the principles of quantum mechanics, which provide security guarantees fundamentally distinct from classical cryptography. The primary

security features of BB84 are as follows:

- **No-Cloning Theorem:** The impossibility of perfectly copying an unknown quantum state ensures that an eavesdropper cannot replicate qubits without introducing detectable disturbances.
- **Uncertainty Principle:** Measurements in incompatible bases (rectilinear vs. diagonal) disturb the quantum state, making eavesdropping evident through increased error rates.
- **Error Detection:** By comparing a subset of their measurement results, Alice and Bob can estimate the error rate introduced during transmission. If this error rate exceeds a predefined threshold, they infer the presence of an eavesdropper and abort the protocol.

The protocol provides information-theoretic security, meaning it is secure against any adversary, regardless of their computational power, as long as the error rate is below a critical threshold.

3.2 Performance Analysis

Since Bob randomly and uniformly chooses the bases in which he measures, the proportion of bases he correctly guesses will be approximately 50% when a large number of qubits are used. Of these 50%, Bob uses a portion (e.g., one-third) of the bits for reconciliation with Alice to detect potential eavesdropping. If the error rate is low enough that it does not suggest eavesdropping but can be explained by random physical inaccuracies, then $\frac{2}{3} \cdot \frac{1}{2} = \frac{1}{3}$ of the qubits will remain for the key. Therefore, the probability of obtaining a valid key after key reconciliation strongly depends on the error rate caused by physical inaccuracies and the algorithm used for key reconciliation. The Hamming encoding variant we selected becomes ineffective when multiple errors occur simultaneously. With a large number of qubits and an increased probability of multiple errors occurring at once, this algorithm thus has a higher likelihood of failure. Hence, there is a trade-off concerning the number of qubits. However, we assume the error rate due to inaccuracies is low, which allows the procedure to most likely produce a valid key.

3.3 Security Analysis

In the case where Eve listens to the channel, she must choose a basis in which to perform the measurement for each qubit. Since Alice randomly and independently selects the bases with equal probability, Eve has a 50% chance of guessing incorrectly for each qubit, regardless of the strategy she chooses. The qubits measured with the wrong basis have a 50% probability of experiencing a bit flip. Since if Bob has chosen the wrong basis, the measurement result is random anyway, eavesdropping detection is of no help. Therefore, the cases of interest are those in which Bob chose the correct basis and Eve chose the wrong one (50% of the qubits that Bob measured correctly). In total, we expect that 25% of the bits measured correctly by Bob will not match Alice's bits. The probability that no bit flip is detected by Bob drops below 5% for just 11 qubits ($1 - (0.75^{11}) \approx 0.958$). Depending on the error rate caused by physical inaccuracies, the threshold for bit flips may need to be adjusted slightly. However, the probability that eavesdropping remains undetected still exponentially approaches zero.

4 Key reconciliation

The key reconciliation process ensures that Alice and Bob share an identical secret key, even in the presence of noise. After Alice and Bob have exchanged qubits and measured them in their chosen bases, they publicly compare a subset of their measurement results. This subset, known as the *public key*, allows Alice and Bob to identify which bits in their respective keys correspond to matching measurement outcomes. In the lecture, various quantum algorithms were discussed in which noise from physical errors is eliminated by simply repeating the algorithm until the results achieve statistical significance. However, this approach is not feasible in the case of this protocol, as repeatedly sending qubits compromises security. If a qubit is sent again and measured in the same basis, but the measured bit shows a different outcome, it is highly likely that the basis in which it was measured is incorrect. Thus, an eavesdropper (Eve) could gain information about the basis used by Alice with very few measurements, without having affected all the qubits through measurement. Moreover, as Eve's ability to guess the basis increases, the qubits are less likely to be disturbed by measurements, making it more difficult to detect the eavesdropper. Therefore, an alternative method is needed to ensure that Alice and Bob share exactly the same key:
First, the keys are encoded using Hamming encoding, which adds error-correcting parity bits. Then, errors are detected and corrected using the Hamming decoding algorithm. The corrected keys are compared, and if they match, Alice and Bob can proceed to generate the final, shared secret key. This process ensures that both parties share a key with high probability, even if some bits have been altered by noise or interception.

5 Discussion

One of the most significant features of BB84 is its ability to detect eavesdropping attempts. Any attempt by an eavesdropper, Eve, to intercept and measure the qubits would disturb the quantum states, causing errors in the key. By comparing a subset of their final key over the public channel, Alice and Bob can detect the presence of such errors and, if the error rate is high enough, abort the protocol. This feature provides a strong security guarantee.
Despite its robustness, the protocol faces practical challenges in real-world implementations. Imperfect detectors, noisy channels, and the need for reliable error correction mechanisms can introduce vulnerabilities and limit the distance over which the protocol can be applied.
Overall, BB84 remains a cornerstone of quantum cryptography, demonstrating the power of quantum mechanics in securing communication.

References

1. Bennett, C. & Brassard, G. Quantum Cryptography: Public Key Distribution and Coin Tossing. *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing* (1984).
2. Hoffmann, F. *Quantum-Cryptography* `https://github.com/mikeagn/CEC2013`.